THE EMPEROR’S OLD ARMOR

Bob Blakley
blakley@vnet.ibm.com

The traditional computer security model is built around a “reference monitor”, supported by hardware protection mechanisms, which enforces administratively defined security policies. The reference monitor’s software is assumed to be of high reliability and integrity. The reference monitor is supplemented by strong cryptography for those unfortunate moments when our data must venture outside the cozy confines of its safe haven.

This model’s analogies are mostly military: the image is that of an *information fortress*, with walls, guards, interior compartments, and a defending army. When you approach the information fortress’s outer wall (“security perimeter”), you present your “password” to the guardian of the gate. The fortress’s defensive garrison (“access control” facilities) protect your “confidential data” until you want to send it out of the “security perimeter”, perhaps through a “firewall”, at which point you use a code (but only in your home country -- because cryptography is a “munition”!) The system’s strong walls and trustworthy gate guards (“integrity features of the Trusted Computing Base”) protect it against the introduction of “Trojan Horses” and “logic bombs”.

The information fortress model was designed for (and in) a world in which computers were expensive, solitary, heavy, and rare. But that world is long gone. Information fortresses are not protecting today’s information much more effectively than Europe’s magnificent physical fortresses are protecting today’s national borders.

The state of computer security is dismal. The same exposures keep recurring; we make no practically useful progress on the hard problems of integrity, assurance, policy, and interoperability; and we are less and less able to adapt the fortress model to new technologies as they arise. Computers are rapidly getting smaller, cheaper, and more richly connected. More and more data resides on machines incapable of meaningful physical security (for example, laptop computers and “personal digital assistants”) and designed -- by economic necessity -- with no strong logical security. Even the relatively few remaining information fortresses have thrown open their gates to Ethernet, ISDN, and fiber connections. At the other end of those connections lies the worldwide Internet, on which, as Steve Bellovin has observed, “There Be Dragons”.

Technologies more disruptive than the Internet loom on the horizon; object-orientation blurs the distinction between data and code, robbing us of one of our most powerful integrity tools (hardware-enforced memory protection). At th .
e same time object orientation encourages us to “reuse” code written by others -- in some cases without benefit of access to the source text of the code we reuse. “Intelligent Agent” architectures invite us to execute other peoples’ code on our systems and to write our own code and send it out to make its way in the world without benefit of our oversight. These agents are not distinguishable from programs which we describe as “viruses” today.

The software industry is in general not keeping up with the escalating threat; most modern software is designed without any thought given to security up-front. The Internet, OMG CORBA, the Worldwide Web, and most Personal Computer operating systems are examples of major components of the worldwide software infrastructure into which security is currently being retrofitted.

The Information Fortress model is based on three principles; the security community’s dirty little secret is that all three of these principles rest on infirm foundations:

1. Policy

Policy scales poorly in every dimension. As the number of subjects authorized to use the system, the number of objects managed by the system, and semantic complexity of operations provided by the system increase, the policy administrator's job quickly spirals out of intellectual control.

2. System integrity and the reference monitor

"System integrity" assures that the security policy of a system cannot be bypassed. The US National Computer Security Center defines "integrity" as follows [NC88]:

> "sound, unimpaired, or perfect condition"

This sets the bar pretty high. But perfection really is the standard, because any hole in the wall of the fortress will let the enemy in.

Implementing a high-integrity system is prohibitively costly and difficult.

3. Secrecy

The fortress model depends heavily on secrecy. The security community has long recognized the problems associated with secrecy and has shrunk the secrecy perimeter to exclude everything except cryptographic keys; this has been formalized as Kerchoff's principle: "security is in the keys", which is intended to mean that if the keys remain confidential, the system is secure. But decades of experience with the problems of passwords and crypto key management suggest that a more accurate formulation might be "insecurity is in the keys"

The simple problem with secrets is that people are not good at keeping them. But there are also complicated problems. It is not always clear, for example, what information constitutes a secret, or what information will reveal it to a particular person.

The central proposition of the paper, therefore, is:

> No viable secure system design can be based on the principles of Policy, Integrity, and Secrecy, because in the modern world Integrity and Secrecy are not achievable and Policy is not manageable.

This is why computer security is starting to fail - and why it will continue to fail until it is re-built on new foundations. The paper urges a search for these new foundations, and suggests some guiding principles:

- Assume low integrity.
- You can't keep a secret.
- Security should be inherent, not imposed.
- Policy is evidence that security is imposed.
- Identity is a side-effect of policy (don't depend on it; don't authenticate it).
- Trust is is evidence that security is imposed (trust nothing and no one).
- Ease of use should be proportional to the probability that use is harmless.
- Make the user ask forgiveness, not permission.
- Plan for emergence.
- Privacy is not secrecy.
- Protection is not control.
- Security is not: confidentiality, integrity, availability.
- Good enough is good enough. Perfect is too good.
- Evolve!